Mobile Robotics: Assignment 1 Report

Mohsin Mustafa - **20161131**, Shyamgopal Karthik - **20161221**

August 23, 2019

Contents

1 LiDAR-Camera Fusion

1.1 Homogenous Matrix Form

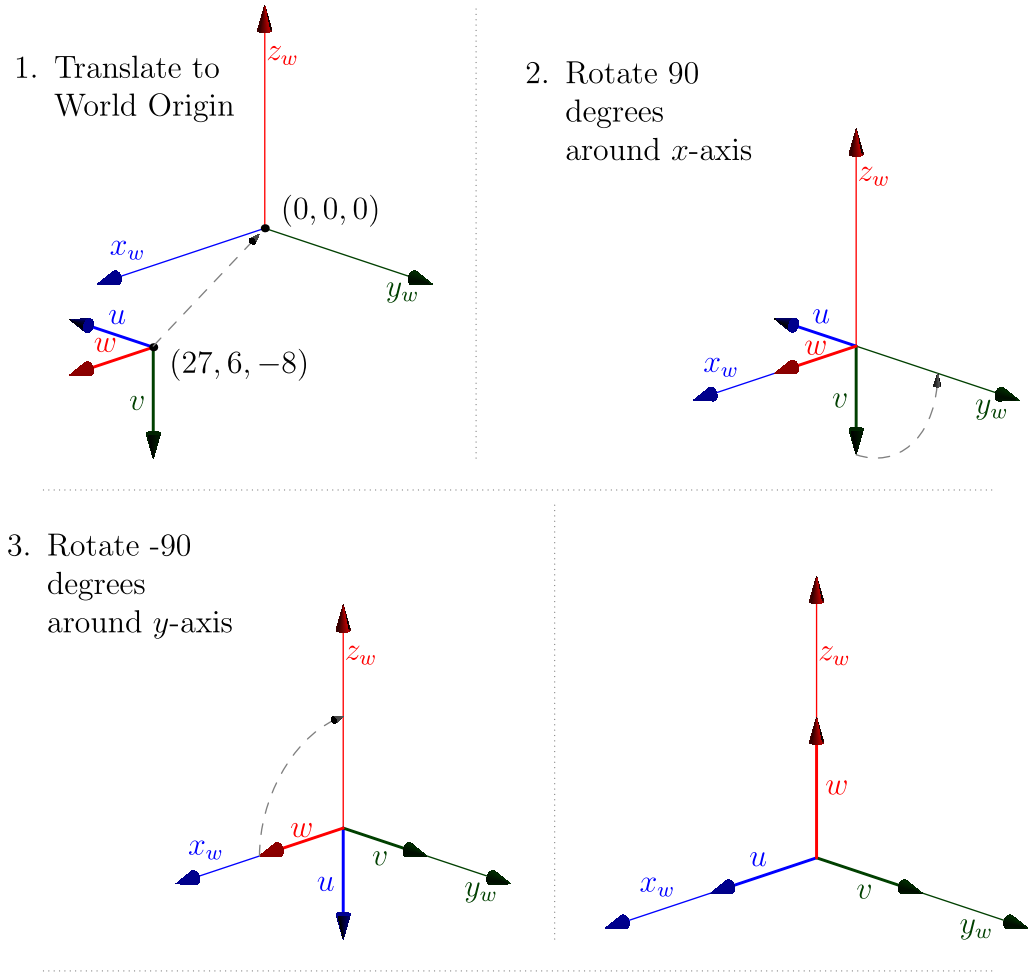


Figure 1: Transforming the camera frame to the world frame. Performing this transformation on the LiDAR points will give the points w.r.t. the camera frame.

Using the above steps, we can find the transformation matrix $\mathbf{M}$ as:

$$\begin{aligned}\mathbf{M} &= R_y(-90) \times R_x(90) \times T(-0.27, -0.06, 0.08) \\ &= \begin{pmatrix} 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & -0.27 \\ 0 & 1 & 0 & -0.06 \\ 0 & 0 & 1 & 0.08 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\ &= \begin{pmatrix} 0 & -1 & 0 & 0.06 \\ 0 & 0 & -1 & -0.08 \\ 1 & 0 & 0 & -0.27 \\ 0 & 0 & 0 & 1 \end{pmatrix}\end{aligned}$$

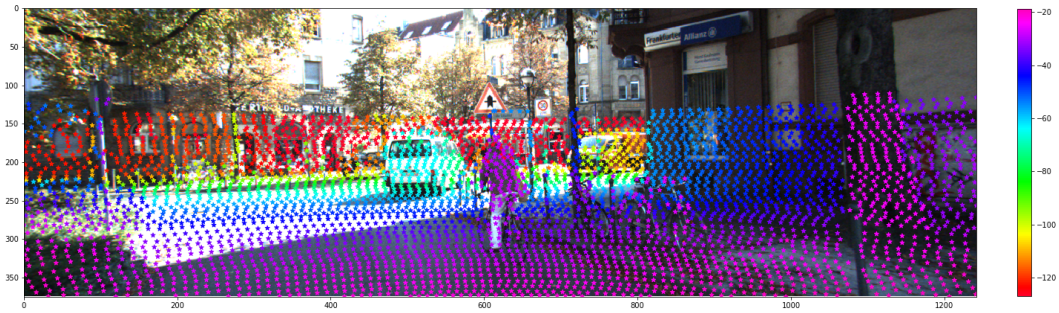
1.2 XYZ Euler Angles—Translation Form

Using the XYZ Euler Angle transformations as shown on the next page, a point $\mathbf{p}$ (world frame) can be represented in the local camera frame as $\mathbf{p}''''$ as follows:

$$\begin{aligned}\mathbf{p}'''' &= \text{Trans}^{-1}(-6, 8, 27)\mathbf{p}''' \\ &= \text{Trans}(6, -8, -27)\text{Rot}_x^{-1}(-90°)\mathbf{p}'' \\ &= \text{Trans}(6, -8, -27)\text{Rot}_x(90°)\text{Rot}_y^{-1}(0°)\mathbf{p}' \\ &= \text{Trans}(6, -8, -27)\text{Rot}_x(90°)\text{Rot}_y(0°)\text{Rot}_z^{-1}(-90°)\mathbf{p} \\ &= \text{Trans}(6, -8, -27)\text{Rot}_x(90°)\text{Rot}_y(0°)\text{Rot}_z(90°)\mathbf{p}\end{aligned}$$

1.3 Image Plane Projection

To project the LiDAR points onto the image plane, the following steps are performed. First the transformation matrix is computed as shown above ($\mathbf{M}$). Then the projection matrix is formed by $\mathbf{P} = \mathbf{K} \times \mathbf{M}$, where $\mathbf{K}$, the calibration matrix, is provided. The scaling parameter is set to one by dividing all elements of $\mathbf{P}$ by $\mathbf{P}_{3,4}$. $\mathbf{P}$ is multiplied with every LiDAR point X as $x = \mathbf{P}X$. The depth is available as x_3. The image plane coordinate values are obtained as $(x_1/x_3, x_2/x_3)$. The depth map is shown here.



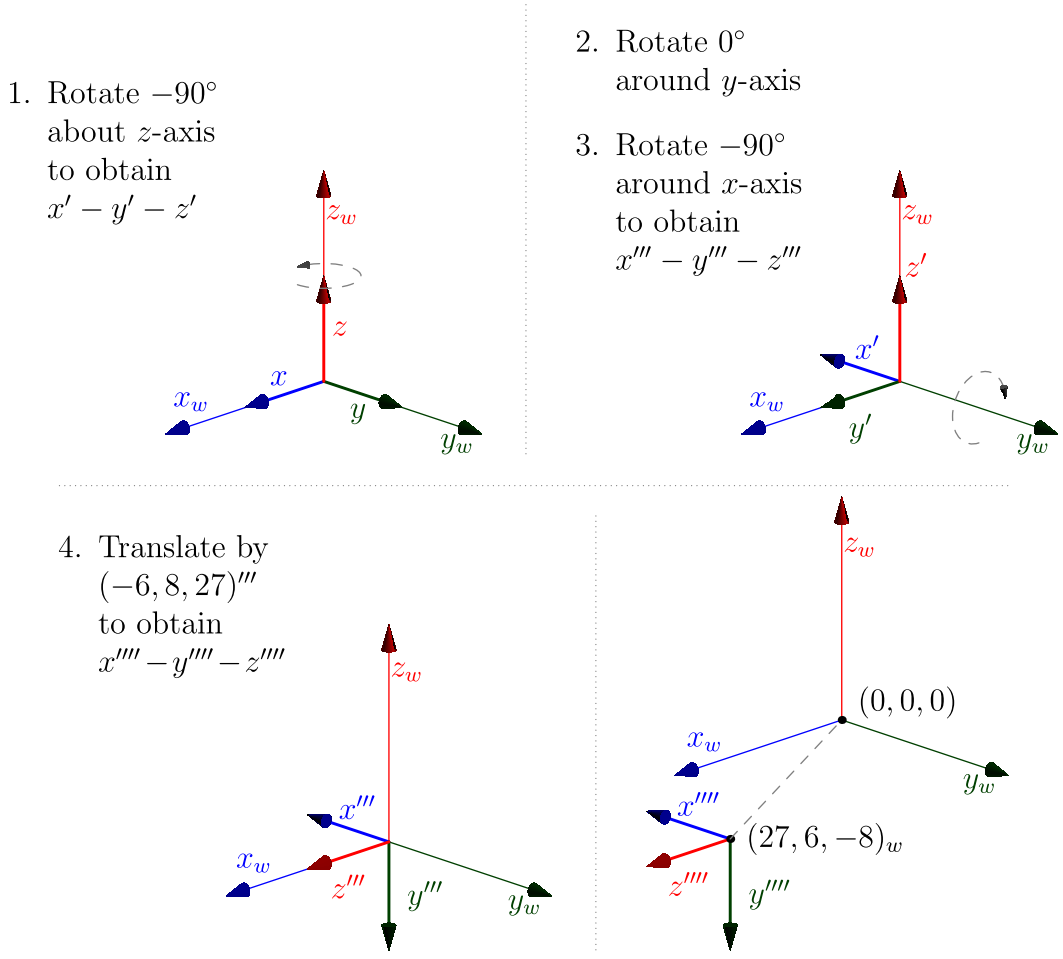


Figure 2: Transforming from the world frame to the camera frame using ZYX Euler Angle-Translation transformations.

2 Drawing 3D Bounding Boxes

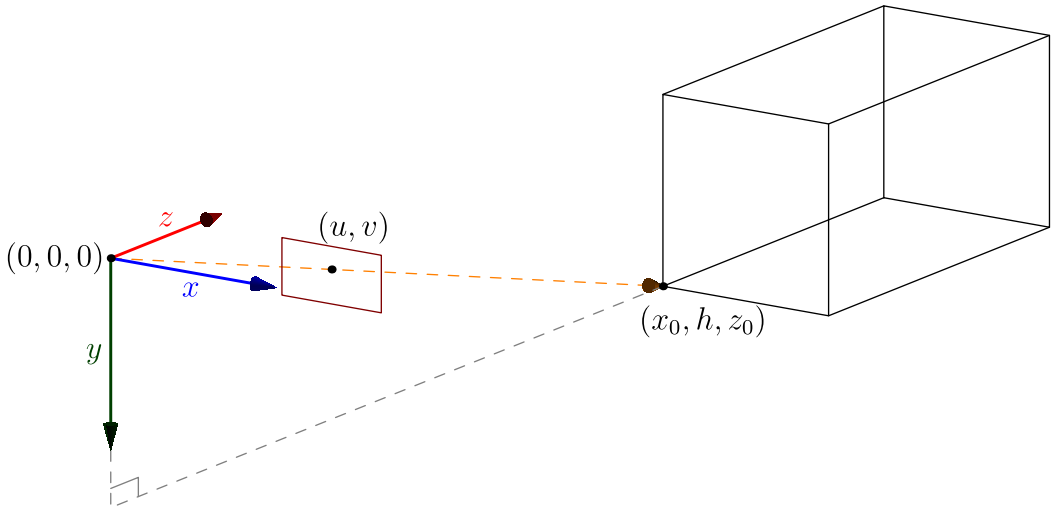


Figure 3: The axis taken for the calculations is centered at the camera. The brown rectangle is the image plane.

The camera origin is h (1.65m in the question) metres above the ground. As shown in the above figure, we first take a point (bottom-left corner of the car) on the image. This point is assumed to be on the ground plane; hence we know the height of this point in terms of the camera's y-coordinate (h). We know that

$$\lambda \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \mathbf{K}_{3\text{x}3} \begin{pmatrix} x_0 \\ h \\ z_0 \end{pmatrix}$$

$$\implies \lambda \mathbf{K}_{3\text{x}3}^{-1} \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} x_0 \\ h \\ z_0 \end{pmatrix}$$

Now $\mathbf{K}_{3\text{x}3}^{-1}\mathbf{u}$ is a vector which passes through (x_0, h, z_0) (orange ray in the above figure). To get λ, we know that $\left[\lambda \mathbf{K}_{3\text{x}3}^{-1}\mathbf{u}\right]_2$ must be equal to h. Hence we divide by $\left[\lambda \mathbf{K}_{3\text{x}3}^{-1}\mathbf{u}\right]_2$ and then multiply by h.

$$\lambda = \frac{h}{\left[\mathbf{K}_{3\text{x}3}^{-1}\mathbf{u}\right]_2}$$

$$\implies \begin{pmatrix} x_0 \\ h \\ z_0 \end{pmatrix} = \left(\frac{h}{\left[\mathbf{K}_{3\text{x}3}^{-1}\mathbf{u}\right]_2} \right) \mathbf{K}_{3\text{x}3}^{-1}\mathbf{u} = \frac{h\mathbf{K}_{3\text{x}3}^{-1}\mathbf{u}}{\left[\mathbf{K}_{3\text{x}3}^{-1}\mathbf{u}\right]_2}$$

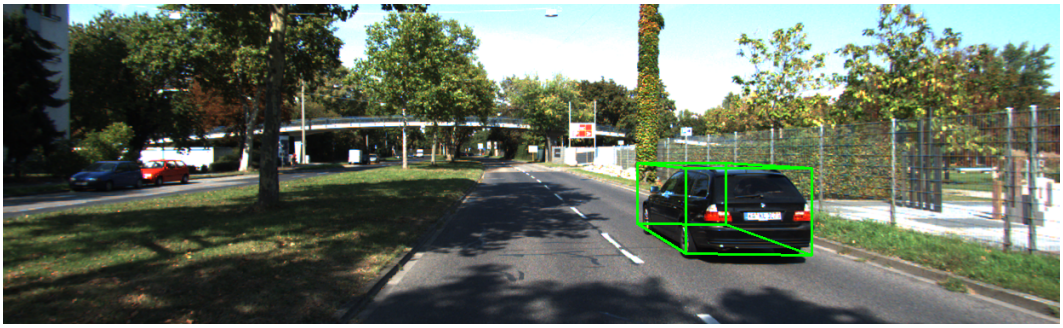


Figure 4: 3D bounding box wireframe projected onto the image plane.

The second element of a 3D vector is its projection onto the y-axis, *i.e*, $\mathbf{v}_2 = \hat{\mathbf{j}}^T\mathbf{v}$. Finally,

$$\begin{pmatrix} x_0 \\ h \\ z_0 \end{pmatrix} = \frac{h\mathbf{K}_{3\text{x}3}^{-1}\mathbf{u}}{\hat{\mathbf{j}}^T\mathbf{K}_{3\text{x}3}^{-1}\mathbf{u}}$$

After computing $[x_0, h, z_0]$, we can find the 3D coordinates of the other seven corners of the bounding box of the car with the dimensions given in the question. To project back all these points to the image plane, we calculate

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \mathbf{K}_{3\text{x}4} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

3 How do AprilTags Work

3.1 Obtaining Homography Matrix

The goal is to find the homography matrix H given the scene points $\mathbf{X}_i$ and the projections $\mathbf{x}_i$. We use multiple point pairs for calculating H. The i^{th} scene point is denoted as $\mathbf{X}_1 = (\mathbf{X}_{1_x}, \mathbf{X}_{1_y}, 1)$ ($\mathbf{X}_{1_z}$ is not taken as we assume all points to be on the same z-axis plane). The i^{th} image point is denoted by (u_i, v_i). The following is the DLT algorithm to solve for these unknown variables:

1. Set up the matrix M and vector v (to solve for) as

$$M = \begin{bmatrix} \mathbf{X}_1^T & \mathbf{0}^T & -u_1\mathbf{X}_{1x} & -u_1\mathbf{X}_{1y} & -u_1 \\ \mathbf{0}^T & \mathbf{X}_1^T & -v_1\mathbf{X}_{1x} & -v_1\mathbf{X}_{1y} & -v_1 \\ \mathbf{X}_2^T & \mathbf{0}^T & -u_2\mathbf{X}_{2x} & -u_2\mathbf{X}_{2y} & -u_2 \\ \mathbf{0}^T & \mathbf{X}_2^T & -v_2\mathbf{X}_{2x} & -v_2\mathbf{X}_{2y} & -v_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}_{2N\text{x}9} \qquad v = \begin{pmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \boldsymbol{\lambda} \end{pmatrix}_{9\text{x}1}$$

2. Due to noise in M, we aim to find v so as to solve for $\min_{||v||^2=1} ||Mv||^2$. Compute the SVD of M: $M = USV^T$.

3. The solution is the eigenvector with least eigenvalue, which is in the last column of V.

The eigenvector is reshaped into a 3×3 matrix to get the homography matrix H. The obtained H is:

$$H = \begin{pmatrix} -0.00183399 & -0.00020696 & 0.00022125 \\ 0.0003863 & -0.00193324 & 0.00025981 \\ 0.00017918 & 0.00129796 & -0.00110334 \end{pmatrix}$$

3.2 Homography Matrix Decomposition (*bonus*)

Using the given K and calculated H, we next aim to decompose H to get R and T. Let $K^{-1}H = (h'_1, h'_2, h'_3)$; we need an orthogonal rotational matrix R that is closest to $(h'_1, h'_2, h'_1 \times h'_2)$:

$$\arg\min_R ||R - (h'_1, h'_2, h'_1 \times h'_2)||^2$$

We find this R by using the SVD of $(h'_1, h'_2, h'_1 \times h'_2) = USV^T$. Then we get R as (the diagonal matrix ensures that $\det(R) = 0$):

$$R = U \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \det\left(UV^T\right) \end{pmatrix} V^T$$

The R we thus obtained is:

$$R = \begin{pmatrix} -0.97595478 & -0.10186357 & 0.19270726 \\ 0.19122635 & -0.82444202 & 0.5326611 \\ 0.10461721 & 0.55670385 & 0.82409712 \end{pmatrix}$$

Figure 5: The scene points projected onto the image plane using the calculated H.

The translation is obtained as $T = h'_3/||h'_1||$:

$$T = \begin{pmatrix} 0.11751003 \\ 0.13799209 \\ -0.58601634 \end{pmatrix}$$